

# *Ambilight alternatív megvilágítási technika Arduino által vezérelve & screen capturing alkalmazásfejlesztés DirectX API felhasználásával*

Tallódi László  
Informatika Kar  
Debreceni Egyetem, Műszaki Kar  
Debrecen, Magyarország  
laszlo.talodi91@gmail.com

Erdei Timotei István  
Mechatronikai Tanszék  
Debreceni Egyetem, Műszaki Kar  
Debrecen, Magyarország  
timoteierdei@eng.unideb.hu

**Absztrakt**—2004-ben, Philips által kifejlesztett Ambilight televíziók megjelenése nem hozott átütő sikert. 2010-ig mindösszesen négymillió ilyen tévét tudtak értékesíteni. A 3D technológiához hasonlóan az Ambilight szerű megvilágítási technológiák is igen megosztóaknak bizonyultak [1]. Projektben egy Ambilight szerű, saját megvilágítási technika került megvalósításra, amihez az inspirációt az Ambilight technológia adta. A cél, hogy asztali számítógépes rendszereken is könnyen alkalmazható legyen és stabil-megbízható működést lehessen vele elérni, aminek fő platformja Windows operációs rendszer. A projekt megvalósítása során megismerésre került a DirectX API által biztosított, úgynevezett „screen capturing” technológia, valamint az Arduino és intelligens, LED-enként vezérelhető LED-szalag programozási ismerete is.

**Kulcsszavak**—*Arduino; Ambilight; Windows; DirectX; FastLED; Screen Capturing*

## I. BEVEZETŐ

A különböző eszközök és azok alkalmazása az oktatásban, illetve fejlesztési célból, komoly szakaértelmet kíván. A Debreceni Egyetem, Épületmechanikai Kutatóközpontjában a jelenkor technológiai találmányai meg, melyekkel új, innovatív rendszereket tervezünk [11][12].

Az Ericsson cég által végzett, éves TV és médiahasználatról szóló jelentése (ConsumerLab TV & Media Report) [2] alapján 2010-2016-os időszakban a tradicionális TV nézés mértéke folyamatosan csökkent. A trendek változásával egyre több időt töltünk úgynevezett Video On Demand (röviden VoD) szolgáltatások használatával, amellyel a lineáris TV adások nem tudnak versenyezni.

Mindemellát az Ambilight-tal felszerelt TV készülékek sikerének elmaradása részben indokolható a pluszszolgáltatással járó magasabb árral [1]. A projekt tervezése során így törekedtem annak minél kisebb mértékű anyagi teherrel járó megvalósítására.

Az Ambilight jellegű megvilágítási technológia lényege a képernyőn megjelenő képből (általában a kijelző kávéjának vízszintes és függőleges oldalainak mentén) mintavételezési

pontokon szinképet elemez és az a káva hátulján lévő LED lemásolja, ezáltal tágítja ki a néző képi érzékelését. Ehhez szükség van egy vezérlő egységre is.

Erre a szerep betöltésére egy Arduino board lett kiválasztva. Az Arduino a világ egyik vezető Open-Source platformja, amely magába foglalja mind a szoftveres fejlesztőkörnyezetét (IDE), mind pedig a hardvert. Segítségével a felhasználók működő elektronikai prototípusokat tudnak létrehozni, akár önállóan működő, akár számítógép által vezérelve. A cég számos szoftveres és hardveres eszközt kínál. Az első Arduino eszközt 2005-ben mutatták be. Az Arduino típusok két nagy csoportra oszthatók. A fő fejlesztő paneleket „Board” névvel illetik, míg a bővítő lapok elnevezése: „Shield”. Az évek alatt több különböző modell jelent meg, ezek közös jellemzője, hogy Atmel mikrovezérlőket alkalmaznak és a programozási nyelvük azonos [3].

## II. TERVEZÉSI SZEMPONTOK

A tervezés legfontosabb szempontja, annak a technológia eljárásnak a megtalálása és használatának beépítése, amellyel az intelligens LED-ek vezérlése legalább olyan kis válaszidővel valósuljon meg, amely még nem zavaró az emberi szem számára. A projekt során további szempont volt, hogy a LED szalag és vezérlésének elhelyezése, úgy valósuljon meg, hogy minél kisebb helyet foglaljon, és ne zavarja a monitor későbbi fali konzolra történő telepítését se [13].

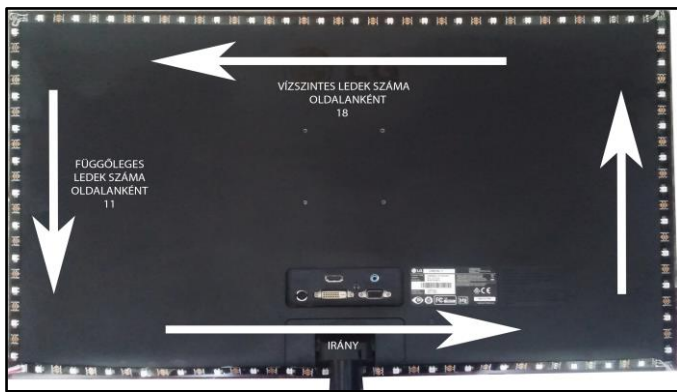
A Windows operációs rendszeren futó szoftver tekintetében pedig, hogy olyan kommunikációs port kerüljön használatra, amely minden számítógépen megtalálható, így nem szükséges módosítás a program alap működését tekintve.

A számítógép monitorjával szemben ülve magát a vezérelt fényforrás közvetlen nem látható ezért szükség van egy olyan fényforrásra, amely képes elegendő fényt kibocsátani úgy, hogy a monitor mögött elhelyezett elég nagy felületű tárgyról (pl.: fal) visszaverődő fény ne is domináljon, de ne is legyen lényegesen kisebb erejű, mint maga a monitor által

kibocsájtott fény. Természetesen a fényforrás vezérlését is meg kell oldani úgy, hogy a vezérlő jelet ne kelljen ismételni az adott monitor hátulján elhelyezett, viszonylagos nagyobb mennyiségű fényforrás esetében sem [14]. A tervezés során pozitív jelentőségű szempont, hogy könnyen felszerelhető legyen.

### III. LÉTREHOZOTT RENDSZER FELÉPÍTÉSE

Egyszerűsített működését illetően az Ambilight három részegység együttműködésével történik. Fordított sorrendben tekintve szükség van egy fényforrásra, amely jelen esetben egy olyan LED-szalagot jelent, amely képes RGB szín keverésére, könnyen telepíthető a monitor hátoldalán lévő borításra, és vezérlése megoldható Arduino felhasználásával.



1. ábra: LED-szalag elhelyezési iránya, száma

A LED-szalag pontosan WS2812B típusú, úgynevezett intelligens LED (Light Emitting Diode) szalag került kiválasztásra, amelyen 5x5 mm-es 5050-es SMD RGB LED egybeépítve található a LED vezérlő IC-jével. Emellett a LED-szalag sűrűsége 30 LED/méter [4].

#### 1. TÁBLÁZAT WS2812 FONTOSABB PARAMÉTEREI

Paraméter	Jel	Határérték	Egység
Tápfeszültség (vezérlő)	VCC	+6,0 – 7,0	V
Tápfeszültség (LED)	VDD	+6,0 – 7,0	V
Vezérlő bemeneti jelszint	DI	- 0,5 – VDD +0,5	V
Áramfelvétel (LED-enként / szín)	I	0 – 20	mA
Adatsebesség	-	800	kbps

Az 1-es ábrán látható monitor méreteiből adódóan vízszintes oldalak mentén 2x18db, míg a függőleges oldalak mentén 2x11db LED került elhelyezésre. Ezeket összeadva következik, hogy 58 db LED vezérlését kell megoldani.

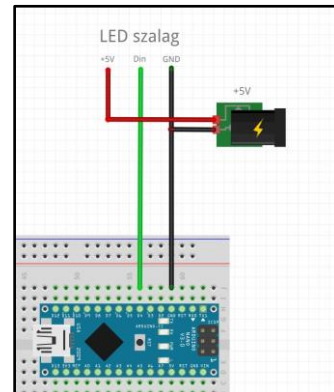
Az 1-es táblázat adataiból látható, hogy egy LED energia felhasználása RGB színekkel számolva összesen 60mA teljes fényerősségen. A fentiekből adódóan a maximális energiafelhasználást a következő táblázat taglalja.

#### 2. TÁBLÁZAT ENERGIAFELHASZNÁLÁS

	Érték	Egység
LED-ek száma	58	db
Áramfelvétel (Összes LED-et tekintve)	3480	mA
Áramfelvétel (Mikrokontroller)	50	mA

Áramfelvétel (összesen)	3530	mA
-------------------------	------	----

A LED-szalag energiafelhasználását egy külső, 5V-os adapterrel került megvalósításra, míg a Nano USB porton kapja a működéséhez szükséges feszültséget.



2. ábra: LED-szalag és Arduino Nano összekötése

Második nagyobb részegység a LED-szalagot vezérlő mikrokontroller. Erre a feladatra egy Arduino Nano (továbbiakban Nano) lett kiválasztva, amely tartalmaz egy ATmega328-as (16MHz-es) mikrokontrollert, valamint Micro USB-n keresztül programozható. Méretéből (18x45mm) adódóan ideális választás a tervezési szempontok részben leírtak alapján [5].

### IV. ARDUINO NANO PROGRAMOZÁSA

A Nano programozása az Arduino ingyenesen elérhető (1.6.12 verziószámú) IDE-jében történt [6]. Az Arduino programozása egy egyszerűsített C nyelven történik. Nem feltétlenül szükséges érteni a mikrokontroller működési elvét, elég az IDE menüpontjai között kiválasztani az alappanel fajtáját és az arra beépített mikrokontroller típusát. A programozáshoz felhasználásra került egy harmadik forrásból származó könyvtár: FastLed, amely támogatja a WS2812B lapkakészletét [7].

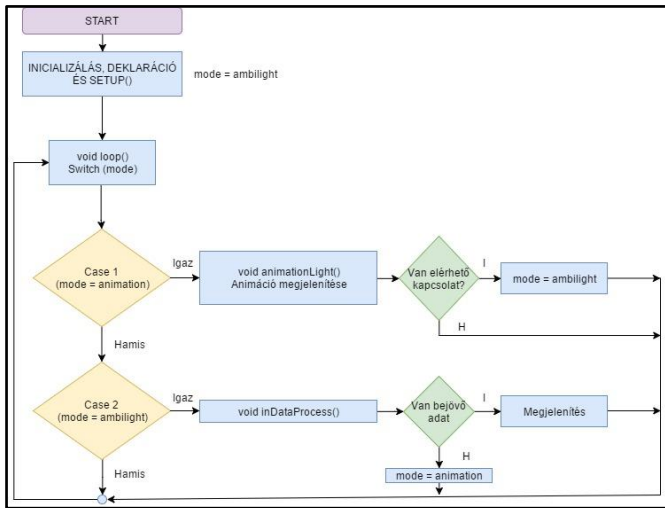
```
byte buffer[NUM_BYTES]; // Maximális bufferméret ledek száma * 3
// ANIMÁCIÓ
CRGBPalette16 currentPalette = RainbowColors_p;
TBlendType currentBlending = LINEARBLEND;
uint8_t startIndex = 0;

void setup()
{
  FastLED.clear(true);
  FastLED.addLeds<WS2812B, DATA_PIN, GRB>(leds, NUM_LEDS);
  Serial.begin(BAUDRATE);
  FastLED.setBrightness(BRIGHTNESS);
}

void loop()
{
  switch (mode)
  {
    case MODE_ANIMATION:
      fillLedsFromPaletteColors();
      break;

    case MODE_AMBILIGHT:
      processIncomingData();
      break;
  }
}
```

3. ábra: Részlet a Nano-n futó program forráskódjából



4. ábra: Részlet a Nano-n futó program forráskódjából

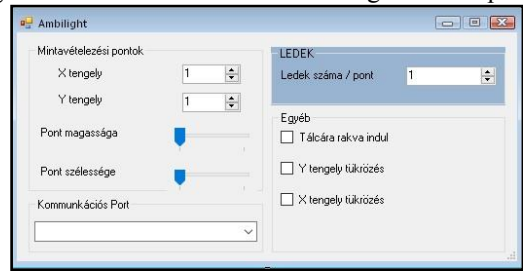
A 4. ábrában látható, hogy kettő értéket vehet fel a „mode” nevű változó. Ez a későbbi tesztelés elősegítésére került bele. Ha a program a Nano bemenetén nem érzékel jelet vagy bejövö adatot átvált egy úgynevezett animációs módra (a LED-szalagon körülhalad egy szivárványszín), így kizárva, hogy nem a Nano-n futó programmal, energiaellátással van probléma.

#### V. WINDOWS OPERÁCIÓS RENDSZEREN FUTÓ „SCREEN CAPTURING” ALKALMAZÁS FELEPÍTÉSE

Többféle technika ismert a képernyö beolvasására Windows Operációs rendszer alatt. Ezek közül a két legismertebb a GDI és a DirectX API-t felhasználó technika. Ez előbbi a képernyöt egy ablaknak tekinti, és erröl az ablakról pillanatképeket készít. Hátránya, hogy a folyamat rendkívül megterhelö a processzor számára. A DirectX alapú alkalmazás tartalmaz egy puffer-t, amely tárolja az alkalmazás videó memóriájában található tartalmat, ez az úgynevezett „back buffer”. Ezen kívül van megtalálható egy „front buffer” is, amely a képernyöröl tartalmaz információt. A „front buffer”-t felhasználva hozzáférhetünk a különböző Windows operációs rendszeren futó alkalmazások ablakának tartalmához [8].

A screencapturing alkalmazást a Microsoft Visual Studio Community 2015 4.6.01055-ös verziójában lett megvalósítva Microsoft Windows 10 Operációs rendszer neve Microsoft Windows 10 Enterprise N 2015 operációs rendszer alatt [9]. Továbbá felhasználásra került a SlimDX nevü, ingyenes, nyílt forráskodú keretrendszer, elősegítve a könnyebb, .NET-en alapuló, DirectX API-t használó alkalmazások írását [10]. A program C# nyelven íródott és több osztályt tartalmaz. Külön osztály kezeli a mintavételezési pontok beállítását és külön osztály kezeli például az adatok továbbítását a kommunikációs porton. A program kinézetét a Visual Studioban található Form segítségével került megvalósításra. Felépítése egyszerűnek tekinthetö. A működéshez elengedhetetlen paraméterek megadásán kívül

lehetőség van extra funkcióként arra, hogy a program indítása után egyböl tálcára rakva induljon el, illetve a későbbi újra felhasználás érdekében megvalósításra került az X és Y tengely mentén történö tükrözés lehetőségének bekapcsolása.



5. ábra: Végleges program kinénete

```
lock (PontSet.Lock) // nem enged másikat szölet amíg más szölet is van
{
    foreach (mintaPont spot in PontSet.Spots)
    {
        if (spot.BalFent.Poz >= 0
            && spot.JobbFent.Poz >= 0
            && spot.BalLent.Poz >= 0
            && spot.JobbLent.Poz >= 0)
        {
            //SlimDX eljárás position + read négyzetpontként(5 vs
            dataStream.Position = spot.BalFent.Poz;
            dataStream.Read(colorBuffFelBal, 0, 4); // buffer, offset

            dataStream.Position = spot.JobbFent.Poz;
            dataStream.Read(colorBuffFelJobb, 0, 4);

            dataStream.Position = spot.Center.Poz;
            dataStream.Read(colorBuffCenter, 0, 4);

            dataStream.Position = spot.BalLent.Poz;
            dataStream.Read(colorBuffLentBal, 0, 4);

            dataStream.Position = spot.JobbLent.Poz;
            dataStream.Read(colorBuffLentJobb, 0, 4);

            ErtekAtlagolo();
        }
    }
}
```

6. ábra: Részlet a forráskodból

A program esetében a mintavételezési eljárása a következőképpen történik: A program elindítása után megadható, hogy hány pontröl történjen a mintavételezés. A kívánt hatás érdekében ennek meg kell egyeznie a monitor hátulján felhelyezett LED-ek darabszámával. X tengely jelöli a vízszintest, míg az Y a függöleges tengelyt jelenti a káva szélén.



7. ábra: A mintavételezés szemléltetése

A 7. ábrán szemléltetésre került az úgynevezett négyzet alapú mintavételezési pontok elhelyezkedése. Ebben az esetben 15 vízszintes és 8 függöleges négyzet alapján történik az eljárás. A LED-szalag felhelyezési irányát

figyelembe véve az első négyzetből (7. ábra jobb alsó sarkában lévő négyzet) lekérésre kerül 5, a 8. ábrán látható pozícióban lévő mintavételezési helyen (bal felső, jobb felső, középső, bal alsó, jobb alsó) lévő pixel ARGB szerinti színparamétere.



8. ábra: A négyzetben történő mintavételezési eljárás elnagyolt illusztrálása

Ez 4 darab paramétert (alpha, red, green, blue) jelenti, ahol az alpha az áttetszőséget jelenti. Ezután ezek átlagolása történik, majd tárolásra kerül, mint az osztálya egy objektuma. Az összes négyzeten belül megtörténik a mintavételezési eljárás, majd egy külön osztály, amely a soros porton csatlakozott Nano-val való kapcsolattartásért felel, ezen az objektumok paraméterét a kimenő bufferbe eltárolja, majd a megadott és sikeresen csatlakozott kommunikációs porton továbbításra kerül.



9. ábra: Próba, demó felvétel lejátszása

## VI. ÖSSZEZGÉS

A projekt megvalósítása során egy működő Ambilight szerű megvilágítási rendszer került megvalósításra asztali számítógépes környezetben. Szubjektív teszteléssel megállapítható, hogy eltekintve néhány feltáratlan színkeverési, illetve a vélhetően a vezérlés késéséből fakadó hibát, az asztali számítógépes Ambilight szerű kialakítás élvezhetőbbé teszi mind a számítógépes játékot, mind pedig a

filmnézést, azáltal, hogy jelentősen kitágítja a monitor képének érzékelését.

## KÖSZÖNETNYILVÁNÍTÁS

Köszönet illeti a Debreceni Egyetem Mechatronikai Tanszékének oktatóját a rendelkezésre bocsátott segítségért.

A publikáció elkészítését az EFOP-3.6.1-16-2016-00022 számú projekt támogatta. A projekt az Európai Unió támogatásával, az Európai Szociális Alap társfinanszírozásával valósult meg

## HIVATKOZÁSOK

- [1] Elefant, (2017, May 20). [Online]. Available: <https://prohardver.hu/tudastar/ambilight.html>
- [2] Ericsson Consumerlab, (2017, May 20). [Online]. Available: <https://www.ericsson.com/assets/local/networked-society/consumerlab/reports/tv-and-media-2016.pdf>
- [3] Málna PC, (2017, May 20). [Online]. Available: <https://malnapc.hu/arduino/az-arduino-plattform/>
- [4] Worldsemi WS2812 manual, (2017, May 20). [Online]. Available: <http://www.seeedstudio.com/document/pdf/WS2812B%20Datasheet.pdf>
- [5] Arduino, (2017, May 20). [Online]. Available: <https://www.arduino.cc/en/Main/ArduinoBoardNano>
- [6] Arduino, (2017, May 17). [Online]. Available: <https://www.arduino.cc/en/Main/Software>
- [7] Daniel Garcia (2017, May 21). [Online]. Available: <https://github.com/FastLED/FastLED/wiki/Chipset-reference>
- [8] Gopalakrishna Palem, (2017, May 21). [Online]. Available: <https://www.codeproject.com/Articles/5051/Various-methods-for-capturing-the-screen>
- [9] Microsoft (2017, May 05). [Online]. Available: <https://www.visualstudio.com/downloads/>
- [10] SlimDX (2017, May 22). [Online]. Available: <https://slimdx.org/index.php>
- [11] Mizik S, Baranyi P, Korondi P, Sugiyama M, „Virtual Training of Vector Function based Guiding Styles” Buletinul stiintific al universitatii politehnica din timisoara romania seria automatica si calculatoare / scientific bulletin of politechnica university of timisoara transactions on automatic control and computer science.
- [12] Ohashi Toshiyuki, Szemes Peter, Korondi Peter, Hashimoto Hideki, „Nonlinear disturbance compensation for haptic device,” IEEE International Symposium on Industrial Electronics. Konferencia helye, ideje: Bled, Szlovénia, 1999.07.12-1999.07.15. Abu Dhabi: pp. 304-309.
- [13] T. I. Erdei, Zs. Molnár, N. C. Obinna, G. Husi, „Surveillance and Security System in the Building Mechatronics Research Center,” Electrical Engineering and Mechatronics Conference EEMC'16 - 7th-18th-19th March, 2016.
- [14] T. I. Erdei, Zs. Molnár, N. C. Obinna, G. Husi, „Térfigyelő rendszer kialakítása az épületmechatronikai kutatóközpontban,” MTEK MR Engineering Science in the North Eastern Region of Hungary – 25. May-2016.